International Journal of Technology and Design Education (2024) 34:505–530
https://doi.org/10.1007/s10798-023-09838-6

“You give a little bit more love to animals than to robots”: primary pupils’ conceptions of ‘programming’ and programmable artefacts

Cecilia Axell[1] · **Astrid Berg**[1]

Accepted: 5 June 2023 / Published online: 26 June 2023

Abstract

Although digital technology is an important part of young people’s lives, previous research implies that they have a limited understanding of what programming is and its connection to the digital devices they encounter every day. In order to create conditions for meaningful teaching in and about programming in technology education, more knowledge about younger students’ pre-understanding and experiences is needed. In the light of this, the aim of this case study was to explore young pupils’ descriptions of the concept ‘programming’, in connection with being introduced to programming as a teaching content in technology education. The study is based on semi-structured interviews with 16 children in year 1 (7-year-olds) in a primary school in Sweden. In their descriptions of ‘programming’ as an activity, the pupils mainly used technological descriptions—a theory of artificial mind perspective. However, when they talked about the objects with which they associated programming, psychological descriptions—a theory of mind perspective—were more clearly present. Then, a less pronounced distinction between humans and machines was made. Anthropomorphic references were used, such as when the pupils referenced children’s culture such as movies and television programs. However, the term ‘programming’ was difficult for many of the pupils to grasp. They also had difficulty in finding a function for programming, as well as explanations and arguments for why they learn programming in school. The results of this study indicate that these 7-year-old pupils perceive ‘programming’ as something complex. This at the same time as they describe how programmed and programmed artefacts (including AI devices) are highly present in their everyday lives, in their leisure environments, and in school. This mirrors how technology has become an ‘intelligent’ and active agent, rather than a mere tool in their lives—an aspect that teachers may forget to take advantage of.

Keywords Technology education · Primary school · Programming · Theory of mind · Theory of artificial mind

✉ Cecilia Axell
cecilia.axell@liu.se

Astrid Berg
astrid.berg@liu.se

[1] Department of Behavioural Sciences and Learning (IBL), Division of Learning, Aesthetics, Natural Science (LEN), Linköping University, 601 74 Norrköping, Sweden

Introduction

> Programming is everywhere and fundamental to the understanding of a hyper-connected world. Coding is the literacy of today and it helps practice 21st century skills such as problem solving, teamwork and analytical thinking (European Commission, 2018).

Today, the digitalisation of society is a significant element of technological development, and it permeates an increasing part of our daily lives. It is, therefore, an important societal issue, not least from a civic and democratic perspective, since many of today's political and societal decisions are connected to technology (Swedish National Agency for Education, 2022). However, when technology becomes more digitalised, new skills are needed, focusing on increased digital literacy and new ways of thinking. As a result, computer programming/coding is listed as one of the twenty-first century's literacy skills, as it is the foundation of all digital technologies (European Commission, 2018; Lye & Koh, 2014; Nouri et al., 2020; Resnick et al., 2009; Tikva & Tambouris, 2021). A discussion about the role of Computer Science (CS) and programming for all students has also arisen. Today an increasing number of countries have introduced, or are planning to introduce, learning to code/program into their school curriculum (Heintz et al., 2017; Manches & Plowman, 2017; Mertala, 2019; Williamson et al., 2019). In some countries, Computing Teaching (CT) has been introduced as a separate subject (e.g., England), while in others it is integrated with other school subjects, for example by including digital competence and programming as interdisciplinary content through the curriculum (e.g., Finland) (Heintz, et al., 2017). A third way of introducing programming in compulsory school is to include this content in extant school subjects. This is the case in Sweden.

The overall aim of the technology subject in Swedish compulsory schooling is to develop pupils' understanding of technological solutions in everyday life. Due to the digitalisation of society, programming was implemented into the curriculum as a part of digital competence in the school subjects technology and mathematics in 2018 (Cederqvist, 2020; Stigberg & Stigberg, 2020; Swedish National Agency for Education, 2022; Vinnervik, 2022).

Although programming is often emphasised as new content in Swedish elementary school education, it has existed, with different purposes and to varying degrees, since the latter part of the 1960s. An important difference from previous attempts to include this content in schooling is that programming instruction in Sweden today includes all students, from the lower ages in primary school to upper secondary school (Swedish National Agency for Education, 2022; Rolandsson, 2019). However, this is a challenge, since far from all primary teachers are prepared to teach programming (Swedish National Agency for Education, 2019). Additionally, research exploring how this reintroduced content has been deployed in school practice is scarce (Stigberg & Stigberg, 2020). Hence, in Sweden, there is a request for more research on programming in early years education (Kjällander et al., 2016).

Through the teaching of foundational programming concepts, young learners can be introduced to essential ideas connected to the design of many of the digital objects they use and interact with in their everyday lives (Bers, 2008; Sullivan & Bers, 2016). This way of approaching programming content can contribute to an understanding of the technological world of which they are a part (Bers, 2008). Nevertheless, there are few studies investigating young children's conceptions and prior knowledge about today's highly digitalised world. For example, there is little known about children's perceptions of what 'programming' is (Geldreich et al., 2019), as well as the artificial minds of the complex digital and programmed devices by which they are surrounded (Spektor-Precel and Mioduser, 2015a; 2015b). In addition, research on programming as teaching content in compulsory technology education is still very limited in scope.

Since children are growing up surrounded by digital technologies, they start forming their conceptions of these and how they work at an early age. However, pupils' conceptions and prior knowledge of programming can both support and hinder what they are supposed to learn (Cederqvist, 2020; Geldreich et al., 2019; Rücker & Pinkwart, 2016). The starting point for this study is that it is important for technology teachers to have knowledge of pupils' pre-understanding and conceptions of the subject content (Mawson, 2007, 2010; Milne & Edwards, 2011; Outterside, 1993; Roden, 1995; Siu & Lam, 2005), in this case programming (Geldreich et al., 2019; Manches & Plowman, 2017; Mertala, 2019).Thus, the aim of this case study is to explore young pupils' descriptions of the concept 'programming' in connection with being introduced to programming as teaching content in technology education. The overarching research questions are:

What beliefs and ideas about what 'programming' is do primary pupils express?
How do primary pupils describe the artefacts associated with programming activities?

As a theoretical framework, we use Mitcham's (1994) fourfold dimension of technology in combination with the concepts Theory of Mind (ToM) and Theory of Artificial Mind (ToAM), as defined in Spektor-Precel and Mioducer (2015a; 2015b).

Our study will contribute with research-based knowledge of the development of teaching strategies in and about programming in compulsory school technology.

Background

Digital competence and programming in compulsory technology education in Sweden

In the late 1970s and early 1980s, informatics education, which included teaching about computers as well as learning programming, was established in many countries including Sweden (Heintz et al., 2017; Rolandsson, 2012, 2019; Rolandsson & Skogh, 2014) In the Swedish curriculum for compulsory schooling from 1980 (Lgr 80), CS was introduced in the subject mathematics in years 7–9. The focus in Lgr 80 was the computer, which was highlighted from three different perspectives: democracy, working life, and as a tool in teaching. The results of this effort showed that most of the teaching focused on skills training and simple computer applications (Riis, 1991).

As computers became more user-friendly and the amount of software increased, their use at work, at school and at home rose. Many teachers, however, did not have enough knowledge in the subject, not least in programming. This led to a shift in focus in the curriculum from 1994 (Lpo 94)—from learning *about* computers, to learning *with* computers (Heintz et al., 2017).

In the middle of the 1990s, technology developed further. Wireless networks became more common, which led among other things to CD-ROM-based materials being replaced by interactive and online services. Many teachers and students were given access to their own computers, which made it possible to use IT as a tool in lessons and thus in teaching (Olteanu, 2020).

In 2015, the Swedish government commissioned the National Agency for Education to draw up guidelines for an updated national strategy for the Swedish school system. One part of this work was to revise the curriculum from 2011 (Lgr11) for primary and upper secondary education. *Digital competence* is visible throughout the revised curriculum (2018a), as well as the curriculum that enters into force 2022, but is explicitly formulated in some specific subjects: technology, mathematics and social studies (Heintz et al., 2017; Swedish National Agency for Education, 2018a, 2022).

The writing on digital competence in the curriculum is based on the EU's key competences and the Digitization Commission's descriptions of digital competence. The four aspects of digital competence included in the Swedish curriculum are: to understand how digitalisation affects individuals and the society; to be able to use and understand digital tools and digital media; to have a critical and responsible approach; and to use digital tools to solve problems and put ideas into action (Swedish National Agency for Education, 2018a, 2022).

An important aspect regarding other countries' curricula is that some use the term 'coding', while others use 'programming' as subject content. One possible interpretation is that they are often used synonymously. Manches and Plowman (2017), however, make a distinction between the two terms: coding refers "to the specific skills of inputting instructions using a particular language, such as Java or Scratch, whereas programming reflects the wider design and implementation process of using code to solve particular problems" (p. 193). Their definition of *programming* thus agrees partly with the Swedish National Agency for Education's interpretation of the term. However, the term is broadly defined in the Swedish context and viewed in a wider perspective. It includes concepts of computational thinking as well as digital citizenship (Bocconi et al., 2018; Stigberg & Stigberg, 2020). In the Swedish curriculum, there is an explicit connection between programming and digital competence. For example, in a commentary document to the revised curriculum from the National Agency for Education (2018b), it is made clear that the focus is on programming—not on coding skills—as a pedagogical tool, problem-solving process and impact on society, since the aim is for pupils to develop a general understanding of programming as well as its effects on society (Bocconi et al., 2018; Heintz et al., 2017; Stigberg & Stigberg, 2020; Swedish National Agency for Education, 2018b).

The purpose of programming as subject content is to give pupils the opportunity to orientate themselves in an increasingly digitalised everyday life, as well as to prepare them for further studies and working life. This includes a general understanding of programming and rules that form the basis of digital systems, i.e., a basic understanding of digitalisation in society in which algorithms, code and software are included (Heintz et al., 2017; Mannila, 2017; Swedish National Agency for Education, 2022).

In the current syllabus of technology for compulsory schooling (2022), it is stated that pupils shall develop an understanding of how computers and networks work. For example, in years 1–3, pupils are supposed to learn about what computers are used for, their parts for input, output and storage, and common artefacts that are controlled by programming such as household appliances and smartphones. In years 4–6, core content involves learning some basic component parts of a computer, their functions (like processor and working memory), and how computers are controlled by computer programs and how they can be connected in networks. Regarding programming, it is stated that in years 1–3 pupils should be given opportunities to control objects with programming, and in years 4–6 they control their own constructions or other objects with programming (Swedish National Agency for Education, 2022).

Research about children's conceptions of coding/programming, computers and the Internet

Many studies examining children's perceptions/conceptions or notions of what 'programming' is do not examine it as an isolated phenomenon, but rather one connected to the Internet, computers, robots, programmable toys, or other digital artefacts (see e.g. Cederqvist, 2020; Kuperman & Mioduser, 2012; Mertala, 2019. Additionally, and as mentioned above, the terms 'programming' and 'coding' are often used synonymously, even if a distinction is sometimes made (see e.g., Campbell & Walsh, 2017; Lye & Koh, 2014; Manches & Plowman, 2017).

Research about children's conceptions of digital technology can be traced as far back as the 1960s, and the first studies exploring children's understanding of the Internet are from the early 2000s (Mertala, 2019; Rücker & Pinkwart, 2016). However, in the wake of accelerating digitalisation, the ways we interact with digital tools like computers have changed dramatically, and have little in common with 20 years ago. Today's children are surrounded by complex digital devices, which likely has an impact on how they form their conceptions of them (e.g., Chaudron, 2015).

Robertson et al. (2017) use the results from a study done in 1987, and make a comparison between children's conceptions about computers 30 years ago and today to see how/if their ideas have changed as digital technology has advanced. In the study, 18 primary school children in Scotland (aged 5–8 years) were interviewed. Robertson et al. (2017) note that even if the pupils had good knowledge of how to use computers, they had difficulty in explaining how a computer works and can be programmed. Some pupils considered the behaviour of computers as being human—i.e. computers can 'think'; while others were of the view that since computers do not have a brain, they cannot think. There were also pupils stating that computers can only follow orders. One conclusion drawn from the study is that it is important to give pupils opportunities to develop an understanding of the difference between how a human mind works and how a machine—in this case, a computer – works. This knowledge may contribute to being able to draw conclusions about strengths, potential risks and limitations of the technologies that surround them (Robertson et al. (2017).

Kuperman and Mioduser (2012) is another example of a study investigating children's explanations of the behaviour of programmed artefacts—in this case robots with adaptive behaviour. When analysing whether kindergarten children (5 to 6-year-olds) used anthropomorphic or technological language when explaining how programmable artefacts work, the researchers found that the more complex the task, the more the children's descriptions shifted from a psychological (using anthropomorphism) to a technological perspective. One possible explanation, according to Kuperman and Mioduser (2012), is that more complex tasks require analysis and interpretation based on the actual function of the programmed artefact. Consequently, the children's focus had to shift from the observable to the underlying causes of the robot's behaviour, which requires the use of analysis and interpretation skills focusing on the artefact's functional components (Kuperman & Midouser, 2012).

Similarly to Kuperman and Mioduser (2012), Diethelm et al. (2017) emphasise the importance of both pupils and teachers having a language, i.e. terms, to guide them in activities with digital devices. In their study, they investigated how German pupils arranged, categorised and distinguished 23 different terms related to the digital world. Through observations, the researchers established that the pupils often avoided technical terms and instead used terms such as "thing" and "it". A quarter of the eight to ten-year-old pupils did not draw any connection between robots and other digital artefacts, and some stated that robots have nothing to do with the Internet or computer science. Diethelm et al. (2017) conclude that pupils come to school bringing many experiences of the digital world which could be of use in the classroom. Thus, it is important that teachers are aware of their pupils' preconceptions about digital artefacts like robots and computers. This knowledge could support teachers when designing their lessons (Diethelm et al., 2017).

Mertala (2019) explored 5 to 7-year-old Finnish children's conceptions of computers, code, and the Internet. The empirical data consisted of drawings produced by 65 children, and interviews with the children. Most of the children had no idea how code and programming are related to computers. The meaning of the terms 'programming' and 'coding' were not familiar to them, and almost half of the group could not answer the question of what programming and/or coding are. If the terms 'code' or 'coding' were used, they were mainly associated with pin codes and passwords needed to log into a computer or to unlock touchscreen devices. The terms 'program' and 'programming' were associated with reading manuals or watching television programs. Only three children had some understanding that programming is about giving commands. According to Mertala (2019), a possible explanation for these three children having this understanding is that they had experience from playing with programmable toys or had played coding games. It was also observed that the children rarely came up with the idea of programming by themselves. Even if they had experience of programmable toys like Bee Bots, it was doubtful whether they could discover the connection between programming a toy and the principles of computer programming by themselves. To be able to do so, notes Mertala (2019), they needed guidance from adults. Much of what the children knew was based on what they had observed their parents doing. An important aspect that Mertala (2019) raises is the fact that teachers and parents often see children as "born-savvy technology user[s]" who "just [pick] it up" when it comes to learning about technology. However, the study's findings challenge the myth of "digital natives". Mertala (2019) concludes that since linguistic cues seem to play an important role in children's conceptions of 'code' and 'programming', an investigation of children's preconceptions of the terms is essential for effective teaching.

In line with Mertala (2019), Geldreich et al. (2019) claim that it is important for teachers to have knowledge about what experiences and beliefs their pupils bring to the classroom from their lives outside school, since children's prior knowledge may both support and hinder learning and thus have an impact on what and how they learn. Their study is based on a qualitative content analysis of filmed teacher-student discussions involving 61 German pupils in years 3 and 4 (ages 8–11). The aim was to provide insights into primary pupils' ideas and knowledge about programming. The results showed that children associated both objects and actions with the term 'programming'. The researchers were surprised to find the subcategory 'creating' recurringly present in all discussions. However, the most frequently mentioned objects being programmed were consumer electronics, computers, and games, which Geldreich et al. (2019) note is not surprising, since these are devices with which the children come into contact in their daily lives. Other objects mentioned were cameras and movies. Some children explicitly mentioned animated films, while others seemed to refer to the cameras used during the discussions. In some of the group discussions, the children were given picture cards, and one depicted a road intersection. When asked what they associate 'programming' with, the children mentioned cars and traffic lights as examples of programmed artefacts. In this context, the children explained that "being programmed" meant "the car is programmed to keep a certain distance" (Geldreich et al., 2019).

Like Geldreich et al. (2019), Cederqvist (2020) states that, since many of today's technological solutions are controlled by programming, pupils need to understand programming as a part of technological solutions, for example in mobile phones. Through semi-structured interviews, Cederqvist (2020) investigated how 11 to 12-year-old pupils understand programmed technological solutions. All pupils in the study had experience of using different programming materials like LEGO Mindstorm, Mircro:bit and LEGO WeDo. Prepared contexts were used to facilitate pupils in answering the questions, which consisted of Micro:bit constructions and programmed technological solutions familiar from daily life: a car key, a remote control and a digital thermometer. The study's results indicate that visible parts (code in the Micro:bit constructions or buttons in everyday technology) were more clearly understood, while invisible parts like a flow of information were too abstract and harder to grasp. It was also difficult for the pupils to discern that there is a code involved in everyday technological solutions—even if they expressed that the objects are programmed. Cederqvist (2020) concludes that concrete programming materials like Micro:bit do not necessarily develop pupils' understanding of programmed technological solutions. To support pupils' understanding, it is important to visualize and pay attention to parts difficult to discern, like the logic of the code, how components are controlled by codes and the information flow that determines the code (Cederqvist, 2020).

To summarise, although digital technology is an important part of children's lives, previous research implies that children have a limited understanding of what programming is and its connection the digital devices they encounter every day. This indicates that in order to create conditions for meaningful teaching in and about programming in technology education, more knowledge about younger students' pre-understanding and experiences is needed.

Theoretical framework

Mitcham's fourfold dimension of technology

To reach an understanding of what technology can be and how it can be described, we can turn to the philosophy of technology. Since programming is technology, to categorize the pupils' descriptions of what 'programming' is, we have chosen Mitcham's (1994) fourfold dimensions of technology as a theoretical framework. A reason for using Mitcham's model is that it is broad and can include many kinds of technology. It is also characterised by being independent of time (Svenningsson, 2020).

Mitcham (1994) identifies four different ways of describing technology: as *object*, *activity*, *knowledge*, and *volition*. With technology as an *object*, Mitcham refers to technology as artefacts, i.e., physical, or abstract objects used in technological activities, or which are a result of technological activities. Based on Mitcham's description, buildings, cars, bicycles, mobile phones, sewing machines and pencils can be examples of physical objects, while software, computer programmes, the Internet, musical works etc. can be examples of abstract objects. Technology as *activity* refers to technological activities such as creating, designing and manufacturing objects, but also the use of technological objects or processes. Hence, activity can refer to craftsmanship, inventing, constructing, operating, and maintaining technological artefacts. Mitcham's dimension *volition* is based on the perception that technology is a part of human will or intentions, i.e., there is always a motive, or several, for a technological activity. Mitcham describes technology as *knowledge* as knowledge and skills humans use when creating, operating, describing, and explaining technological objects. In this study, we have chosen to also include *pre-experiences* in the aspect of technology as knowledge.

In this study, the fourfold dimensions described above are used to categorise the pupils' descriptions of 'programming' as a concept.

Theory of mind (ToM) and theory of artificial mind (ToAM)

A novice understanding of the meaning of 'programming' may involve understanding what a computational device may, or may not, 'understand' in relation to a human (e.g., Robertson et al., 2017). This can be formulated as a capability to discern differences between the human mind and a machine's mind. Spektor-Precel & Mioduser (2015a, 2015b) describe this in terms of an understanding of Theory of Mind (ToM) and Theory of Artificial Mind (ToAM). ToM refers to a psychological perspective, an ability to describe other people's mental states that explain their behaviors. It is thus about having awareness and understanding that other people can hold feelings, desires, beliefs, intentions, and emotions which are different from one's own, and that their decisions can be based on other desires and beliefs than our own. In the context of the present study, this perspective can, for example, be expressed using anthropomorphic language when describing objects, their behaviour and functions. ToAM, on the other hand, refers to an understanding that machines behave according to what humans have programmed them to do, or according to direct operation.

In the analysis of the empirical data, we identify what in the pupils' descriptions can be interpreted as an understanding of ToM and ToAM.

Methodology

Participants and data collection

The study is based on semi-structured interviews with 16 children in year 1 (7-year-olds) in a Swedish primary school in a medium-sized Swedish city. The pupils were interviewed in pairs. The interviews were audio-recorded and later transcribed. Each interview lasted approximately 25 min.

The Swedish Research Council's research ethics principles in the humanistic-social scientific field (2017) were followed. The participating teacher and the parents of the participating children were informed in a letter about the purpose and implementation of the study. It was clearly formulated that the participation was voluntary, and the participants had the right to withdraw at any time without giving reason. They were also informed as to how data were to be handled and reported, and that confidentiality would be preserved which included a pseudonymisation of their names. The pupils' consent was handled by their parents, and the consent was given in writing. The data was stored and protected in accordance with the General Data Protection Regulation (GDPR).

The interviews were carried out in connection with the pupils being introduced to programming as teaching content in technology education. The teacher had planned the introductory lesson without the researchers' involvement. The focus of the lesson was the difference between a robot and a human, as well as that programming being about providing clear and accurate instructions.

Data analysis

The semi-structured interviews were analysed using a qualitive thematic analysis, which is a method where written or verbal communication is analysed with a focus on similarities as well as differences. This is a non-linear, reflective, hermeneutic process, which involves a recurring moving back and forward between the entire data, and initially identified categories/codes. The interpretation process resulted in several identified themes. A theme can be described as capturing something important in relation to the research questions (see e.g. Braun & Clarke, 2006; Kiger & Varpio, 2020). Thus, the focus was not how many of the pupils expressed something specific based on the aim and research questions of the study (frequency), but rather the variation in their answers. The analysis process was carried out in eight steps, see Table 1.

Table 1 The eight steps in the analysis process

Step 1	The analysis process started with a transcription of the interviews. This was followed by a process of orientation of the raw data, i.e., a reading and re-reading of the transcriptions to acquire a good grasp of the overall content. The aim of this process was to identify important aspects in the data in relation to the study's research questions
Step 2	Excerpts that represented pupil's different descriptions of the phenomenon, i.e., programming, were examined. These descriptions were coded in different colours. For example, when being asked what they think about when they hear the term 'programming,' Pupil A answered "[*I'm thinking of*] *some kind of electrical stuff*", and Pupil B answered, "*that you make a kind of robot*". These two answers were coded with different colours, since one statement referred to electrified objects in general, while the other had a clear connection to robots as objects
Step 3	In the search for potential themes, the coded data (coloured quotes) were examined, and 10 different initial themes were identified and named
Step 4	Re-readings of the transcripts and deep analyses of the pupils' statements led to a review of the coded answers as well as merging and renaming of the initial themes. For example, themes like "household support" and "human needs" were condensed into a final theme named *Programming is a part of human's daily life*. In this process, five final themes were defined
Step 5	In this step, the identified themes were categorised based on Mitcham's (1994) four dimensions of technology: object, activity, knowledge and volition. As already mentioned, based on the pupils' statements, the category *technology as knowledge* was widened to also include the pupils' references to *pre-experiences*
Step 6	The pupils' statements were analysed based on Spektor-Precel and Mioduser's (2015a, 2015b) descriptions of ToM and ToAM. This step was thus about identifying whether they made a distinction between the human mind and a machine's mind in their statements about programming
Step 7	The results of the analysis were compiled, and a description of the findings was made
Step 8	The final step was to change the pupils' names to pseudonyms in order to protect participant identities

Results of the data analysis

The thematic analysis ended up with five themes corresponding to five different ways the pupils described what programming is: (1) *Programming is robots and/or to program a robot*, (2) *Programming is about building and/or constructing something*, (3) *Programming is a part of daily life*, (4) *Programming is for education and/or a future profession,* and (5) *Programming is about toys, play and children's popular culture.*

Based on Mitcham's (1994) fourfold dimensions of 'technology', the phenomenon described was also categorized into four categories: (1) Programming as *objects/artefacts,* (2) Programming as *activities,* (3) Programming as *volition,* and (4) *Knowledge* and *pre-experiences* of programming.

The five themes constructed from the empirical data and the distribution of the four identified categories based on Mitcham's (1994) dimensions of technology over these five themes, are presented in Table 2.

Programming is about robots (generally)

The most frequent aspect of 'programming' the pupils highlighted during the interviews is that programming is about *robots* as physical objects.

Researcher:	[…] what is the first thing [...] you think of when I say the word programming?
Maria and Jana:	Robots! (In chorus)
Researcher:	Anything else?
Maria and Jana:	Ehh…
Researcher:	It is often the case that when you hear a word, it pops up almost like pictures in your head…
Maria:	I'm just thinking of a robot.
Jana:	Me too…
Researcher:	Is there anything associated with robots then … that you think of?
Maria:	Batteries …
Researcher:	Yes, batteries … And you then? (Turns to Jana)
Jana:	Cables …
Researcher:	Mmm…
Jana:	Cords …

Maria and Jana agree that robots have the strongest connection to programming. However, when the researcher asks them if there is anything further, based on robots, that they are thinking about, their associations are with electronics. One possible interpretation is that the pupils see batteries, cables and cords as parts of a robot, i.e. concrete and visible objects. The objects with which the pupils associate the term "robot" are based on a general meaning of the word. However, robots in educational settings (like Bee Bots) are frequently mentioned.

Anna and Martin also conclude that robots are the first thing they think of when they hear the word 'programming'. Martin says, "to program a robot" and Anna refers to children's TV programmes with robots. They continue to discuss movies they have seen (on television) with robots which look like humans. They both express that it is scary when it is not possible to determine whether it is a robot or a human being:

Table 2 Description of the seven themes (row 1) and the distribution of the four categories based on Mitcham's (1994) fourfold dimensions of technology (row 2–5)

Themes programming is about …	Objects/Artefacts	Activities	Volition	Knowledge/Pre-experiences
Robots (generally)	Robots Androids Bee bots	To program a robot To help robots To control a robot To make a robot to understand	To not have to do boring things To do things right To do things in the right order To save time To do things more correctly To do things faster To control a robot/the machine To teach a robot something To help robots (with something)	Visited a relative's job Programmed Bee Bots in preschool and preschool class
Building and/or constructing something	Cogwheels Houses Factories Castles	To build things To build a robot To "dig a castle" To construct something To repair something To create something complex To build a car		Made a robot dog of toilet rolls (at home)
Daily life /Human needs, household support	Robot lawn mowers Robot vacuum cleaners Door codes (alarms) Electrical gadgets TV, remote controls Cars Batteries Internet Mobile phones Apps Power cords Electric wires, cables Machines Batteries	A robot can: Tie shoestrings Cook Iron clothes Sew Dress you Go and get water Make breakfast Make the bed Change sheets (in the bed) Clean one's room Program a house Program a mobile phone Build a car	Friendship (robot as a friend) Robots can help us Robot should not look like humans Make time for important things Get more attention from adults Collaboration (with friends) Robot can replace us	Used codes (door alarms) Saw relatives' lawn mower Programmed "a little machine" called "Boxer" with an app in a mobile phone

Table 2 (continued)

Themes programming is about …	Objects/Artefacts	Activities	Volition	Knowledge/Pre-experiences
Education/ Future professions		Doing the homework Helping with homework Doing maths tasks Controlling a robot on a track Learning about robots Filming	To teach robots (something) To learn maths To get a good job To work with programming To work as a teacher To work as a fire fighter To work as a baker To work as a construction worker To get good grades (in programming) To learn for upper secondary school	Built robots in preschool class Used Bee Bots in maths class
Toys/Play Children's popular culture	Lego robots Robot animals Remote control toys TV programmes Movies Scratch Junior program (Computer) games	To program a toy Play, e.g. "To program each other" To use Junior Scratch Computer games		Visited a toy store (robots) Had robot toys at home Used toys with remote controls 'Programmed' each other (a play) Watched children's TV programmes Watched movies (WALL-E) Used Scratch Junior at home

Martin: On the children's channel there is a program about robots. One has a large claw… and then there is one that looks just like a human.

…

Anna: [They are] almost like robots but they can do more themselves. They tell them too [what to do], but they are androids.
Researcher: 'Android'… what is that then?
Martin: It's a kind of robot!
Anna: Yes, but… not really a robot.

…

Martin: It is a robot that manages itself.

The researcher points out that this is an interesting aspect since "some people try to create robots that are human like. But you think the difference should be clear…?" Anna and Martin confirm their view and emphasise that it is important to be able to distinguish robots from humans. Anna concludes that she would not like to have a robot which is too human like, and Martin agrees.

The most common *activity* the pupils express is 'to programme a robot'. Other activities mentioned were 'to make a robot to understand what you want it to do, 'to get help from a robot', 'to help robots' and 'to help the robot if it crashes and gets sad'. The descriptions of how humans need to 'help' robots, and the recurring use of the pronoun 'he' when talking about robots, are examples of how the pupils anthropomorphise the robots, i.e. use humanoid characterizations. The activities described also include a reverse perspective: that programming also means 'helping' or 'teaching' the robot something, or 'comforting' it if it crashes and becomes sad. The *knowledge* aspect was rarely present in the interviews included in this category—only through statements like 'to teach the robot something'. Regarding the *volition* of programming robots, the pupils explain that the robots can do what humans find boring, but also that robots can do things more precisely/correctly, faster and in the right order, and can thus be used by humans to save time.

Most of the pupils express that they have *pre-experiences* from programming a Bee Bot in, preschool class (6-year-olds) or during school lessons in mathematics. Jana says that she has done "different things" in preschool, including "programming a robot" that is actually a friend pretending to be a robot, and that she had also "built a robot" Maria associates programming with giving a Bee Bot instructions: "I have also been doing programming… in [preschool school class] we had a robot for which you would press arrows and then 'go', and then it would go forward two or three steps …". Maria has also "programmed a friend" at her "previous school." Sonja focuses on the buttons on the Bee Bot when talking about programming:

Sonja: It looked like a bumblebee […] which was lying on the ground … there were buttons on [it]. […] I pressed the forward button three times […] and then I pressed the back button [...] then it goes forward three times and then it goes backwards once. And if you want to make it stop doing that, then you press the button in the middle, and [that is the same button] you press when you are going start [the robot].

Wilma also refers to previous experiences and activities from preschool class and notes that there are buttons on the Bee Bot and that you have to *steer* it. She uses the word “steer” instead of “programme”, and can give an exact and correct description of how to program the robot:

Wilma: There were buttons like this, one forward one backwards […] and then you had to try to do, it was like a road like this […] and then you had to try to steer it to the […] road…

Two pupils, Maria and Alma, describe how they have programmed a robot when visiting a relative’s workplace:

Maria: I have visited my grandfather, who works in a factory, and he had a robot I was allowed to play with … he danced … he jumped up and down.”

Alma’s experience, on the other hand, was with programming a robot to make it perform tasks which resembled its regular function:

Alma: “Yes, [I have programmed] at my mother’s job ... It was very difficult but the tenth time I tried it worked ... [the robot] was to fetch water ... but he took a small glass that was very heavy. He did not understand that he should pick up the big glasses because they are lighter ... Finally, we removed the small glasses, so he had to take the big one instead.

Except the references to robots in children’s culture (movies, TV programmes etc.) the pupils’ discussions can be described as taking their starting point from a technological perspective, since they concern how humans can use and program robots to perform tasks in order to solve human problems or to make work easier. The contexts vary: programming robots at home, working or studying at school.

Programming is about building and/or constructing

Another frequent *activity* that pupils mentioned during the interviews, which we found surprising, was the idea that ‘programming’ is about building something—not only robots. All pupils, except Max, connected the word ‘programming’ with activity ‘building’. Wilma explained that she thinks of ‘building something’ and ‘building a car’ when she hears the word ‘programming’: and Asta thinks about building ‘some electrical appliances’. Mikael is not explicit, but says that programming is connected to creative and complicated processes:

Mikael: Maybe you mend something or create [...] something complex. Something which is more difficult to do. You connect things which are ‘tangled’, and you have to think.

One possible interpretation of Mikael’s description is that he sees ‘programming’ as connected to problem solving.

For some pupils, 'programming' is the equivalent of creating or building a robot. For example, Jana explains her pre-experience of programming in preschool as: "we tried a lot of things… we tried to build different robots from different materials".

The connection to building or creating something, no matter what it is, is also made by Maria, who states that she often "does things with programming at home […] I made a dog out of toilet paper (rolls)[…]". Maria reaffirms this view once again during the interview, when she associates programming with a profession as a construction worker, something Tom and Mikael also do.

Maria:	We should learn something about programming, if we are to become construction workers […] because then you must think and so on… how to build and so …

In her explanation, Maria uses 'planning to build' as a metaphor for programming.

Why so many of the pupils express that they associate programming with 'building' something is not clear. However, in the descriptions that fall into this theme, programming becomes an activity that involves a process of thinking and creating, but not necessarily connected to robots or digital artefacts.

Learning programming for school and/or future profession

Learning for school as a volition for programming is something Tom and Mikael mention: it is important to learn programming because they will get grades in school when they get older. Tom states that "[If you do not learn] you may get an F in programming. You may get an F minus minus". The same aspect is also in focus for Wilma and Asta, and they refer to future education:

Asta:	Before you go to high school, you might …
Wilma:	… talk about it [programming] …
Asta:	You have been in [year] one [in school] and then maybe […] if you do not know […] maybe you do not know what to say and so, and what about if you get a test […] and then maybe you get it wrong …
Researcher:	Do you get any benefit from [knowing programming] outside of school?
Asta:	Yes … no …
Researcher:	Or is it mostly for school?
Wilma:	It's mostly for school.

What Asta, Wilma, Tom and Mikael say can be interpreted as seeing programming as a skill you only have use for in a school context. This indicates that the programming activities they have been involved in lack a clear context and clear explanations for why they should learn programming.

Another aspect of 'learning for school' that Jana highlights comes when she explains that, in a future profession as a teacher, you need to know programming: "[t]o be able to teach the children […] And to be able to teach their own children too… Yes, a child might start at the same school where his mother works [as a teacher]".

However, as shown in the description of the theme of programming as being about building something, Maria, Tom and Mikael believe that programming is a skill that construction workers need to have in their profession. Similarly, Anna and Martin mention that knowledge in programming is something that they can benefit from in a future context outside of school.

Researcher:	Do you think… when you grow up and start working… do you think you will benefit from being able to programme?
Anna, Martin:	Yes!
Researcher:	In what way? Have you thought about what you want to work with?
Anna:	I want to be a baker!
Researcher:	Aha… and why is it good to be able to programme then?
Anna:	Well, so that you can… use an electric whisk … And then maybe you need a robot to help if you do not have so many [humans] who can work.
Researcher:	Mmm, who can help?
Martin:	Firefighter!
Researcher:	Mmm… and why is it good to be able to programme then… if you become a firefighter?
Martin:	Sometimes… It can actually be *buttons* so that the water comes out instead […].

What Martin says is an additional example of how programming is associated with pressing buttons. Another example of an outside school context and learning for a future profession is Maria's explanation that you need to be able to programme if you are to become a construction worker.

The pupils' answers categorised as 'Learning for school and/or future profession' suggest that they have rather vague perceptions of the usefulness of learning programming. They express they have experience of programming, both in and outside of school contexts. A common experience is making a Bee Bot to move—for example between different math tasks on the floor—or pressing different buttons on a Bee Bot to make it move in different directions. These descriptions can be interpreted as seeing programming as a tool for solving problems, but also being about giving instructions to a digital tool (like a Bee Bot) or performing tasks in a certain order. However, the example of a baker using an electric whisk shows how associations end up far from real programming situations. However, this can also be interpreted as associating programming with pressing buttons to make use of tools (objects) in different professions.

Programming as a part of the daily life: human needs and household support

A fourth theme we identified in the interviews is about *programming in daily life*, and how programming is connected to human needs in the household. Some of the *objects* mentioned within this category are linked to robots—robotic lawn mowers and robotic vacuum cleaners—but also to digital artefacts like doors with codes, the Internet, mobile phones and apps. Other technological objects like doors lights, television, cars, batteries, power cords, cables electric wires and cars were also mentioned in the discussions about using programming in daily life.

A description of a kind of 'servant robot' was common and linked to *activities* like: to program robots to serve you when you are sick, help you to tie your shoes, iron clothes, sew, dress you, go and get you water, make breakfast, make the bed, change sheets in the bed, clean your room, do your maths homework or serve you when you want to listen to music.

Erica:	Ironing … Ironing our clothes […] Cooking food … Sewing … Dressing us … Or just …
Alma:	[…] what is boring… [..] Cleaning up […] what is most boring, all the robots can do.
Researcher:	So, the robots should do that?
Alma:	Yes, [it can] pick up sweets [for us] and stuff like that. Then we do not have to do it … then it brings [us] a whole bag of candy.

According to Alma and Erica, it is possible to program robots to perform chores they find boring themselves, so they have time to do other things that appeal to them more, which can thus be interpreted as the *volition* of the activities. Another *volition* connected to human needs mentioned by the pupils was that a programmed robot can be a friend but also but also do some of the parents' or other relatives' chores so that they, or their siblings, can get more attention from the adults. Erica explains that: "[the robot] could cook so mum would not have to cook […] because she [also] must take care of the baby"—her sibling. Anna says that a robotic lawn mower could help her get more attention from her grandfather:

Anna:	"There are [robot] lawn mowers too ... so we do not have to do it ... [cutting the grass] sounds boring ... My grandfather usually [cuts the grass] and he isn't that fun then ... he walks around with ... what is it called ... headphones ... so he cannot hear much, and when I try to call to him, he does not hear [me]... and it's pretty boring.

Thus, Anna also refers to a human need when she makes the connection between programming a robotic lawn mower and the *volition* to get more time with her grandfather. However, limitations connected to human needs were also mentioned by the pupils:

Erica:	You can, if you have a robot, if you say [..] if you say you can get water [...] yes water.
Alma:	If you have a robot at home that does everything.
Erica:	Yes cleaning, cooking, make breakfast … everything … toasting bread.
Alma:	But it cannot go to the toilet for us.
Researcher:	It cannot …?
Erica:	Well, it can carry us to the bathroom.
Researcher:	It can carry …
Alma:	Although, it must have a huge body then.

Martin and Andrea relate programming to their *pre-experiences* from using door codes (alarms):

Martin:	Programming a house.
Researcher:	Mmmm… and how can you programme a house?
Martin:	When the doors lock themselves sometimes… there are robots that say, 'access forbidden'.

Researcher: Like a little guard… at the door?
Martin: Yes, access forbidden!
Anna: My grandmother and grandfather have one […] you enter a code in the basement when you go in and out… or only when you go in…
Researcher: Instead of using a key?
Anna: Yes… and we have a code like that… and then it's like a robot that says 'wrong code'… it can sound sometimes too [she sings a short melody].
Researcher: Aha… so you can control robots with your voice too? You don't have to press …
Anna: Mmmm… aaaa… That's so cool!

Some of the pupils also associated programming with TV remote controls and robots on the roads—i.e. cars. Wilma, Tom, Andrea and Martin associate programming with lamps. For example, Martin states that "lamps are almost like a robot". A possible explanation for the pupils associating programming with such technology may be, as we already have mentioned, that they see programming as equivalent to pressing buttons and giving instructions to a technical and electric device. Components connected to electric artefacts were also mentioned. Maria associates 'programming' with "batteries" and Jana thinks of "cables" and "cords". Lenny also talks of "fixing" and connecting cables: "if a car breaks down, you can put cables like this".

Digital devices were a kind of electric object the pupils mentioned. For example, Asta says that she thinks about: "programming … something … a [mobile] phone". And later in the interview she adds: "Well … something with the Internet."

Programming is connected to toys, play/games and children's popular culture

The interviews confirm that robots and programming are a part of children's *activities* today, and connected to play and leisure time, i.e. that the *volition* for programming connected to this category is entertainment. Almost all pupils mention toys when they talk about programming, and they make references to *objects* like Lego robots and different kinds of robot animals:

Maria: I have four robots at home that are toys … A bird and a dog and a little guinea pig and a ladybug.
Jana: "I had a [robot] dog … and I also had a robot cat. And they could walk and talk … And … You did not need a remote control for them … you just said [to them what to do] and then it did what it was able to do".

Some of the pupils also refer to a kind of play where a peer pretends to be a robot:

Anna: "In school we played a game where I controlled my friend… go right, go left, go straight ahead, jump. It's great fun!"
Jana: I did that in preschool … programmed [a friend to be] a "robot", and I walked around with him outdoors. We were going to see if he could use the swing and take off and so on ... and in the end it was great fun, because then he was hit by a swing and fell backwards and laughed at the same time!

Another kind of toy frequently mentioned during the interviews is toys controlled by a remote control, such as dolls and radio-controlled cars. Programming is thus described as being about controlling and giving commands to an object without physically touching it.

It is also clear that programming is a natural part of children's popular culture, for example through television programs, movies, children's literature and computer games:

Mikael: I have seen the movie WALL-E. It was a very funny robot.
Andrea: … There is a [children's TV] programme about it… where you can do a few different things with robots and so on…

However, Tom is the only pupil who expresses having pre-experience from programming with a computer program at home:

Tom: I have Scratch at home that I use. Scratch is a programming game so you can make different games ... I try a little, dad helps me sometimes. I'm not making [my own] games yet.

During the interview, it becomes clear that that Tom's father works with programming and that the pupil is frequently engaged in various types of programming activities at home together with his father. Tom also uses correct terms connected to programming activities—like making 'loops'—and he connects programming to "making [computer] games and apps".

Table 3 Examples of pupil descriptions of programming from a ToM and ToAM perspective, respectively

Examples of ToM descriptions of 'programming' (Descriptions in terms of attributing human/animal-like 'minds' to machines—i.e. that they have intentions and feelings)	Examples of ToAM descriptions of 'programming' (Descriptions in terms of instructions from humans to a machine)
"[A robot] cannot eat normal food … it should eat petrol instead … maybe oil? Soup … Oil with metal [pieces] in"	"[In factories] they use robots [to make/build things]"
"If the robot is sad and has hurt itself, someone who is like us real [humans] helps the robot."	"[Programming is about instructions] to make a car."
"[Robots] need to work out just like we do."	"[Programming is about] programming a house […] to make the doors to lock themselves"
"On TV, I have seen a human … who is a robot … it's scary […] you don't know what they are up to"	"[Programming] is about making things in the right order."
"On a children's [TV] channel [I saw] a girl called Eva who is a robot."	"[Programming] is about controlling [a robot]"
"[They are] almost like robots but they can do more on their own. They also tell them (what to do) but they are androids."	"[I have] programmed a robot […] I pressed the forward button twice and then I pressed the backward button"
"[An android] is a robot that manages itself."	"[We] press arrows and then 'go' and then [the robot] moves forward two or three steps
"[In the children's program on TV] there is one [android] who pinches [others] … he hugs very hard!"	[My robot bird] can play back … a recording."
"[I have] a robot dog and a robot cat also. And they can walk and talk."	"Radio-controlled cars can be programmed."
"If you have a robot [toy], you have someone to talk to and play with …"	"You can also program … dolls."
	"I have Scratch at home […] Scratch is a programming game."
	"Programming is about … [computer] games."
	"You can programme the robots [at home] with your mobile phone, and you can download the app."

How 'programming' and the artefacts connected to programming are described

Table 3 below presents examples of how children describe programming and the artefacts they associate with it. The pupils' statements have been categorised here based on an analysis of whether they express the perspective ToM or ToAM.

The analysis of the interviews shows that a ToM perspective on programming as a phenomenon primarily relates to physical *objects*. It is about pupils attributing psychological and human needs to programmed objects such as robots: they need to eat, may become sad, be comforted, injured, or need physical activity. Therefore, most of the statements categorised as expressions of a ToM perspective deal with references in children's culture, such as films and television programs. The discussion about robots and androids in a television program is one such example. However, when the pupils use anthropomorphic language, for example referring to a robot with the pronoun "he", it does not necessarily mean that they attribute a human mind to the object, but can instead be about gender coding of a physical artifact.

Pupils' statements categorised as expressions of ToAM mainly relate to programming as an *activity*, for example giving instructions or controlling various technical objects such as household technology, toys, computer games or robots in industry. All in all, 'programming' from a ToAM perspective is about controlling, problem solving, giving instructions and doing something in the right order.

We also found statements where the pupils use a mix of a ToM perspective and a ToAM perspective, i.e. shifting between the two explanations. For example, this occurred when they made a comparison between humans or animals and programmed objects: "[…] you give a little bit more love to animals than robots"; "[unlike a robot] the dog does not need to be told to walk […] it can walk by itself", and "humans can do as they please, but robots cannot".

Discussion

In this study, we focused on primary pupils' ideas and beliefs about 'programming' and how they describe the artefacts they associate with programming activities. The main purpose was to explore young pupils' descriptions of the concept 'programming' in connection with being introduced to programming as teaching content in technology education.

Based on Mitcham's (1994) fourfold description of what technology can be, all four aspects *object*, *activity*, *knowledge/pre-experiences*, and *volition* were present in the pupils' descriptions of 'programming'. However, *objects* and *activities* were the most common aspects expressed. Robots were the most recurrent *objects* the pupils associated programming with, and 'to build' the most frequently mentioned *activity*.

A common expression of *volition* associated with programming was to programme a robot to do chores the pupils found boring—for example, to do their homework, make the bed or clean the house. These arguments were substantiated with explanations such as that the pupils, their parents or other relatives would then have more time for 'fun' and 'important things'. Another aspect of volition that the pupils mentioned was that a programmed artefact, like a robot, makes things faster and more correct compared to humans. A programmed artifact can also be a friend. However, as stated by one of the pupils, you can give 'a little bit more love' to a real animal than to a robotic animal.

All interviewed pupils expressed that they had *pre-experiences* of using programming, both in their free time and in teaching settings. Their perceptions of 'programming' were highly based on their everyday contact with different kinds of technological artefacts: toys, games, remote controls, codes to door locks and codes connected digital devices, household robots (robotic lawn mowers and robotic vacuum cleaners), and previous educational activities with Bee Bots in preschool class. But, in line with Mertala (2019) and Geldreich et al. (2019), the pupils also had experiences of fictional programmed objects in children's culture, i.e. television programs, movies and computer games.

However, the term 'programming' was difficult for many of the seven-year-olds to grasp and to pronounce. When expressing what they associated 'programming' with, they mainly described it as: "making something", "programming a robot" or "building something". This is in line with Diethelm et al. (2017) and Mertala (2019), who conclude that linguistics seems to play an important role in children's conceptions of terms like 'programming'. For example, Diethelm et al. (2017) found that the pupils in their study lacked a language (terms) which could guide them during activities involving digital devices, and therefore used terms such as "thing" and "it". This may also be an explanation as to why the pupils in our study used anthropomorphic language: they proceeded from what was well known to them to describe something complicated, in this case programming—a phenomenon also highlighted by Kuperman and Mioduser (2012).

Based on their everyday experiences, the pupils in our study anthropomorphised the robots with which they associated programming. This aspect is also confirmed by Kuperman and Mioduser (2012) and Robertson et al. (2017), who note that, due to the complexity of digital artefacts, children often attribute human minds, thoughts, emotions and intentions to them. For example, in Kuperman and Mioduser (2012), 5 to 6-year-olds used psychological (anthropomorphic) language when explaining the behaviour and functions of programmable artefacts, but shifted to technological language when describing the causes of the behaviour of the artefacts.

In the descriptions of 'programming' as an activity (controlling, programming an artefact, using codes, giving instructions, pressing buttons etc.) the pupils mainly used technological descriptions (ToAM). A ToM perspective was more clearly present when they talked about the objects with which they associated programming. Then, a less pronounced distinction between human and machine was made. A ToM perspective is also present when anthropomorphic references were used, i.e., when the pupils referenced movies and TV programs; when two pupils talked about "androids"; and when they discussed robots with the pronoun "he". In the discussion about androids and the importance of robots not being human-like, the pupils also expressed what can be interpreted as critical thinking about the relationship between humans and technology.

The participants gave many examples of programmed artefacts as well as how to use programmed artefacts like robots, or other digital artefacts such as door locks, artefacts controlled by remote controls and so on. However, similarly to Robertson et al. (2017), the participants had difficulty explaining how these artefacts work in relation to programming. The descriptions of how to programme artefacts were often limited to being about 'buttons' or 'pushing buttons'. Even if they expressed that they had experience of programmable toys, like Bee Bots and others, it is doubtful that they could make a connection between their use of the robot toy and the principles of programming from a wider perspective. This is an aspect also identified in Mertala (2019) and Cederqvist (2020). Cederqvist (2020) concludes, for example, that even if pupils are given opportunities to work with concrete programming materials like Micro:bit in technology teaching, this does not necessarily develop their understanding of how programmed technological solutions in their daily life work—e.g. invisible components such as codes that can be difficult to understand (Cederqvist, 2020).

Today's digital technology is complex to a very high degree, and consists of many different interacting components and systems. A full understanding of how they work or are connected is therefore hard to reach. This was also visible in our study. Even if all participants indicated that they had experiences of programming, only one pupil described the connection between programming a digital artefact and its function/performance. The other pupils' explanations were limited to, as mentioned above, being about pressing buttons. Pressing a button became equivalent to programming. Additionally, and in accordance with the results of Diethelm et al. (2017), the pupils did not make connections between robots and other digital artefacts, the Internet or computer science.

The way in which the pupils associated both objects and activities with the term 'programming' is in line with Geldreich et al. (2019) findings. Geldreich et al. (2019) were surprised that the concept "creating" was frequently present in the discussions among the participating pupils. However, in the present study, we found it more remarkable that almost all participants associated 'programming' with 'building', since we found it difficult to fully grasp what their association between programming and 'building something' was. Geldreich et al. (2019) note that a potential explanation could be that the pupils have experiences from participating in maker space activities, since these have increased in recent years. However, the probability that the participants in this study had experience of having participated in such activities is small, since none of them mentioned it. One possible explanation is linked to their pre-experiences of participating in other activities dealing with processes of problem solving, thinking, and creating technological solutions. One of the pupils explained, for example, that she had experiences from "programming" in preschool; an activity where the children constructed and built robots from different kinds of materials. She made a transfer between the experience of building a robot and programming a robot. However, the other pupils' association between building and programming were unclear, since they included all kinds of constructing activities, including constructing objects that are not digital. Additionally, programmed robots, which was the most common object with which the pupils associated programming, are complex digital devices. This may also be one of the explanations as to why the pupils use 'building' to describe what they think about when hearing the term 'programming'—'to build' something is a visible and concrete hands-on activity.

It was clear that the pupils in our study had difficulty in finding a function for programming. Most of them express that they can programme an artefact to do things that humans find boring, such as vacuum cleaning or mowing the lawn. On the other hand, they also ended up suggesting programming a robot to do tasks that they in fact enjoy doing themselves, such as handing out textbooks in the classroom, doing mathematics homework, etc. This contradiction indicates that it may be important to introduce the concept 'programming' in a clear context. As asserted in previous technology education research, if technology is not placed in a context, there is a risk that the connections between artefacts and humans, as well as what kind of implications technology has in a societal context, are disregarded (Axell, 2017, 2018; Mawson, 2007; Siu & Lam, 2005).

When teachers introduce younger students to programming, it is common to link programming to robots. For example, pupils may 'program each other' (one pupil acts a robot' and the other acts as 'programmers'), and in the next step they are tasked with programming small robots, like Bee Bots. Since children are fascinated by robots, this is an easily accessible way into programming. However, although robots are something that interests children, there is also a risk that teachers forget to consider experiences and pre-conceptions that students already have of programming. As highlighted in in previous research, knowledge about pupils' pre-understanding of a subject content is crucial in technology education (Mawson, 2007, 2010; Milne & Edwards, 2011; Outterside, 1993; Roden, 1995; Siu & Lam, 2005). This with reference to the fact that pupils' prior knowledge and pre-experiences can both support and hinder what they learn in and about 'programming' in the technology classroom (Cederqvist, 2020; Geldreich et al., 2019; Mertala, 2019; Rücker & Pinkwart, 2016).

The results of the present study also confirm that technology plays an important role in children's popular culture. Based on references to television programs, movies and games, the pupils describe anthropomorphised technology, including what can be interpreted as illustrations of artificial intelligence (AI). For example, in movies and television programs, robots are portrayed from a psychological perspective (ToM); they have feelings, intentions and can act independently without human intervention. During the interviews, pupils expressed an interest in such popular cultural expressions and the narratives they present. As noted in previous research (e.g., Axell, 2017, 2018), by using narratives that have a strong story line and are of interest to children, technology can be presented in a meaningful context. The inclusion of narratives in teaching about programming may thus contribute to making this content more comprehensible. Narratives can open up discussions on various aspects of programming and digital technology, including the relationship between humans, technology and society, thereby supporting students' development of critical thinking skills.

To sum up, the results of this study indicate that these 7-year-old pupils perceive 'programming' as something complex and abstract which takes place in a 'black box'. This at the same time as they describe how programming and programmed artefacts (including AI devices) are highly present in their everyday lives, in their leisure environments, and in school. Smart home solutions and robotic toys are common, and anthropomorphic and intelligent artefacts are natural elements in the popular culture they meet. This mirrors how technology has become more of an 'intelligent' and active agent, rather than a mere tool in their lives.

Conclusions

Based on the findings in this case study, we conclude that programming as a phenomenon is a natural part of children's daily life. In their descriptions of 'programming' as an activity, the participating pupils mainly used technological descriptions, i.e. a ToAM perspective. A ToM perspective was more present when describing the objects they related programming to.

The pupils thus confirmed that children have many pre-experiences of and ideas about programming which they bring into the technology classroom, an aspect that teachers may forget to take advantage of. Programmed devices and artefacts, including artificial intelligence (AI), are also common elements in the popular culture pupils encounter. However, due to its complexity, it was difficult for the pupils to understand what programming is even at a basic level, since they perceived it as something abstract and difficult to grasp. They also had difficulty in finding explanations and arguments for why they learn about programming in school.

To conclude we find that this study, although being a case study, highlights aspects that may be of value to teachers who are in the process of implementing programming in technology education aiming at younger pupils.

Funding Open access funding provided by Linköping University.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

Axell, C. (2017). Critiquing literature: Children's literature as a learning tool for critical awareness. In P. J. Williams & K. Stables (Eds.), *Critique in design and technology education* (pp. 237–254). Springer.

Axell, C. (2018). Technology and children's literature. In M. de Vries (Ed.), *International handbook of technology education* (pp. 895–911). Springer.

Bers, M. U. (2008). *Blocks, robots and computers: Learning about technology in early childhood*. Teacher's College Press.

Bocconi, S., Chioccariello, A., & Earp, J. (2018). *The Nordic approach to introducing computational thinking and programming in compulsory education*. Report prepared for the Nordic@BETT2018 Steering Group.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Campbell, C., & Walsh, C. (2017). Introducing the 'new' digital literacy of coding in the early years. *Practical Literacy, 22*(3), 10–12.

Cederqvist, A.-M. (2020). Pupils' ways of understanding programmed technological solutions when analysing structure and function. *Education and Information Technologies, 25*(2), 1039–1065. https://doi.org/10.1007/s10639-019-10006-4

Chaudron, S. (2015a). *Young children (0–8) and digital technology: A qualitative exploratory study across seven countries*. Publications Office of the European Union. http://publications.jrc.ec.europa.eu/repository/handle/JRC93239

Diethelm, I., Schneider, N., & Brinda, T. (2017). How pupils classify digital artifacts. In *ACM international conference proceeding series* (pp. 99–1000. https://doi.org/10.1145/3137065.3137079

European Commission (2018). *Coding – the 21st century skill*. https://ec.europa.eu/digital-single-market/en/coding-21st-century-skill

Geldreich, K., Simon, A., & Starke, E. (2019). Which perceptions do primary school children have about programming? In *ACM international conference proceeding series* (pp. 1–7). https://doi.org/10.1145/3361721.3361728

Heintz, F., Mannila, L., Nordén, L-Å., Parnes, P., & Regnell, B. (2017). Introducing programming and digital competence in Swedish K-9 education. In *Informatics in schools: Focus on learning programming: 10th international conference on informatics in schools: Situation, evolution, and perspectives, ISSEP 2017, Helsinki, Finland, November 13–15, 2017, Proceedings* (pp. 117–128). https://doi.org/10.1007/978-3-319-71483-7_10

Kiger, M. E., & Varpio, L. (2020). Thematic analysis of qualitative data: AMEE guide no. 131. *Medical Teacher, 42*(8), 846–854. https://doi.org/10.1080/0142159X.2020.1755030

Kjällander, S., Åkerfeldt, A., & Petersen, P. (2016). Översikt avseende forskning och erfarenheter kring programmering i förskola och grundskola. http://omvarld.blogg.skolverket.se/wp-content/uploads/sites/2/2016/06/oversikt_programmering_i_skolan.pdf
Kuperman, A., & Mioduser, D. (2012). Kindergarten children's perceptions of "Anthropomorphic Artifacts" with adaptive behavior. *Interdisciplinary Journal of E-Learning and Learning Objects, 8*(1), 137–147.
Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012
Manches, L., & Plowman, L. (2017). Computing education in children's early years: A call for debate. *British Journal of Educational Technology, 48*(1), 191–201. https://doi.org/10.1111/bjet.12355
Mannila, L. (2017). *Att undervisa i programmering i skolan – Varför, vad och hur?* Studentlitteratur
Mawson, B. (2007). Factors affecting learning in technology in the early years at school. *International Journal of Technology and Design Education, 17*(3), 253–269.
Mawson, B. (2010). Children's developing understanding of technology. *International Journal of Technology and Design Education, 20*(1), 1–13.
Mertala, P. (2019). Young children's conceptions of computers, code, and the Internet. *International Journal of Child-Computer Interaction, 19*, 56–66.
Milne, L., & Edwards, R. (2011). Young children's view of the technology process: An exploratory study. *International Journal of Technology and Design Education*. https://doi.org/10.1007/s10798-011-9169-1
Mitcham, C. (1994). *Thinking through technology: The path between engineering and philosophy*. University of Chicago Press.
Nouri, J., Zang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry, 11*(1), 1–17. https://doi.org/10.1080/20004508.2019.1627844
Olteanu, C. (2020). *Datorer och programmering i undervisningen – Då och nu*. Modul: Programmering i grundsärskolan. Skolverket. https://larportalen.skolverket.se
Outterside, Y. (1993). The emergence of design ability: The early years. In *IDATER 1993 conference, Loughborough, Loughborough University* (pp. 43–49). http://hdl.handle.net/2134/15
Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60–67. https://doi.org/10.1145/1592761.1592779
Riis, U. (1991). *Skolan och datorn: Huvudrapport och sammanfattning av utvärderingen av treårssatsningen 1988–91 på datateknikanvändning i skolan*. Linköping University.
Robertson, J., Manches, A., & Pain, H. (2017). "It's like a giant brain with a keyboard": Children's understandings about how computers work. *Childhood Education, 93*(4), 338–345.
Roden, C. (1995). Young children's learning strategies in design and technology. In *IDATER* 1995 *conference, Loughborough, Loughborough University* (pp. 21–27). https://hdl.handle.net/2134/1521
Rolandsson, L. (2012). *Changing computer programming education; the dinosaur that survived in school: An explorative study of educational issues based on teachers' beliefs and curriculum development in secondary school*. Lic. thesis. KTH Royal Institute of Technology.
Rolandsson, L., & Skogh, I.-B. (2014). Programming in school: Look back to move forward. *ACM Transactions on Computing Education (TOCE), 14*(2), 1–25. https://doi.org/10.1145/2602487
Rolandsson, L. (2019). Programmering i skolan. *Vägval i Skolans Historia*. https://undervisningshistoria.se/programmering-i-skolan/
Rücker, M. T., & Pinkwart, N. (2016). Review and discussion of children's conceptions of computers. *Journal of Science Education and Technology, 25*(2), 274–283. https://doi.org/10.1007/s10956-015-9592-2
Siu, K. W. M., & Lam, M. S. (2005). Early childhood technology education: A sociocultural perspective. *Early Childhood Education Journal, 32*(6), 353–358.
Spektor-Precel, K., & Mioduser, D. (2015b). The influence of constructing robot's behavior on the development of Theory of Mind (ToM) and Theory of Artificial Mind (ToAM) in young children. In *Proceedings of IDC 2015b: The 14th international conference on interaction design and children* (pp. 311–314). https://doi.org/10.1145/2771839.2771904
Spektor-Precel, K., & Mioduser, D. (2015). 5–7-year-old children's conceptions of behaving artifacts and the influence of constructing their behavior on the development of theory of mind (ToM) and theory of artificial mind (ToAM). *Interdisciplinary Journal of e-Skills and Life Long Learning, 11*, 329–345.

Stigberg, H., & Stigberg, S. (2020). Teaching programming and mathematics in practice: A case study from a swedish primary school. *Policy Futures in Education, 18*(4), 483–496. https://doi.org/10.1177/1478210319894785

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology & Design Education, 26*(1), 3–20. https://doi.org/10.1007/s10798-015-9304-5

Svenningsson, J. (2020). The Mitcham Score: Quantifying students' descriptions of technology. *International Journal of Technology and Design Education, 30*(5), 995–1014. https://doi.org/10.1007/s10798-019-09530-8

Swedish Research Council. (2017). *Good research practice*. Swedish Research Council.

Swedish National Agency for Education (2018b). *Digitalisering i skolan—Möjligheter och utmaningar*. https://www.skolverket.se/publikationer?id=3971

Swedish National Agency for Education (2018a). Curriculum for the compulsory school, preschool class and school-age educare. Lgr11. Revised 2018. https://www.skolverket.se/getFile?file=3984

Swedish National Agency for Education (2019). Digital kompetens i förskola, skola och vuxenutbildning 2018 [Digital competence in preschool, compulsory school and adult education] (Dnr: 2018:1292). https://www.kolverket.se/publikationsserier/rapporter/2019/digital-kompetens-i-forskola-skola-och-vuxenutbildning

Swedish National Agency for Education (2022). *Läroplanen för grundskolan, förskoleklassen och fritidshemmet. Lgr22*. [Curriculum for the Compulsory school, preschool class and school-age educare. Lgr22] https://www.skolverket.se/getFile?file=9718

Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature review. *Computers & Education*. https://doi.org/10.1016/j.compedu.2020.104083

Vinnervik, P. (2022). Implementing programming in school mathematics and technology: Teachers' intrinsic and extrinsic challenges. *International Journal of Technology & Design Education, 32*(1), 213–242. https://doi.org/10.1007/s10798-020-09602-0

Williamson, B., Bergviken Rensfeldt, A., Player-Koro, C., & Selwyn, N. (2019). Education recoded: Policy mobilities in the international 'learning to code' agenda. *Journal of Education Policy, 34*(5), 705–725. https://doi.org/10.1080/02680939.2018.1476735